

REMARKS

By this amendment, claims 1, 3-10, 12-28, 34-41, 43, and 44 are pending, in which claims 2, 11, 29-33, and 42 were previously canceled without prejudice or disclaimer and claims 17-40, 43, and 44 stand withdrawn from consideration pursuant to the provisions of 35 U.S.C. §121, and claims 1, 5, 6, and 12 are currently amended. No new matter is introduced.

The Office Action mailed December 30, 2009 rejected claims 1, 3-10, 12-16, and 41 as obvious under 35 U.S.C. § 103(a) based on *Rao et al.* (US 6,978,453) and *SyncML Meta-Information DTD* (DTD) in view of *Szeto* (US 7,188,143).

The rejection of claims 1, 3-10, 12-16, and 41 under 35 U.S.C. § 103(a) is traversed.

For the reasons set forth at pages 40-53 and 95-103 of the Appeal Brief filed March 30, 2009, and incorporated herein, independent claims 1 and 41, and therefore, claims 3-10 and 12-16, dependent on claim 1, are not obvious within the meaning of 35 U.S.C. § 103(a).

This response will address the specific rebuttal arguments made by the Examiner at pages 2-12 of the Office Action of December 30, 2009.

Independent claim 1 recites, *inter alia*, “receiving at an electronic device a command specifying execution of an **unidentified executable** on first data.” As argued in the Appeal Brief, *Rao et al.* teaches away from the claimed invention because *Rao et al.* specifies the use of a **named** executable. The Examiner argues that the commands in *Rao et al.* are “enhancement commands,” rather than conventional commands, and that “no where in Rao teaches/suggest that Rao’s system cannot operate utilizing an unidentified executable or that Rao’s system operate utilizing only named executable. Additionally, Rao is analogous art, as Rao is in the field of applicant’s endeavor, utilization of the SyncMP technology in a mobile device” (Office Action of December 30, 2009-page 3).

Whether the commands in *Rao et al.* are specified as conventional or “enhancement commands,” the important point is that *Rao et al.* **only** specifies the use of a **named** executable as *Rao et al.* uses commands that are executed by SyncML DM protocol (col. 8, lines 65-67) and, therefore, specifies a named executable to be used, i.e., the Sync ML DM protocol uses exec commands to invoke enhancement commands for the firmware update. If a reference teaches only A, and does not explicitly teach that B cannot also be used, it is improper to ascribe to such a teaching that B can also be used. Similarly, since *Rao et al.* teaches only that a named executable is used, never suggesting anything about using an unidentified executable, it is improper for the Examiner to speculate and invent a suggestion by *Rao et al.* that merely because *Rao et al.* does not explicitly disclose that unidentified executables cannot be used, this is a teaching or suggestion of using such unidentified executables. A prior art reference cannot suggest a claim feature by omitting any reference to it.

Independent claim 1 also recites, *inter alia*, “automatically identifying an executable using the content type determined from the metadata.” Responsive to Applicants’ argument that the proposed combination of references is improper because DTD does not describe any implementation of Sync ML technology, thus failing to suggest how content information can be used to produce any technical effect, the Examiner argues that it would have been obvious to include DTD in *Rao et al.* for the benefit of properly operating in accordance with the Sync ML standard of having meta-data (Office Action-page 4).

Rao et al. uses a system to execute an update process but does not discuss the identification of particular executables for operation on particular content types and does not discuss using metadata from a firmware update for any reason, e.g., determining an executable. *Rao et al.* merely specifies explicit commands to fetch and apply firmware updates. While

DTD is a Document Type Definition specification that defines a set of mark-up that may be used to identify meta-information associated with a Sync ML command, data item, or collection (page 5, section 1), DTD does not disclose “automatically identifying an executable using the content type determined from the metadata” because DTD fails to discuss any executables. Since neither *Rao et al.* nor DTD describes the identification of particular **executables** for operation on particular content types, the combination of *Rao et al.* and DTD cannot disclose, “automatically identifying an executable using the content type determined from the metadata.” *Szeto* is no help in this regard because *Szeto* relates to techniques for controlling an application in an instant messaging (IM) environment. The various types of IM environments described therein redefine functions so that there is a different handling procedure for messages received from different environments. The portion of *Szeto* that the Examiner found most pertinent, viz., col. 12, line 66-col. 13, line 16, relates to an example of a movie trailer being retrieved from a movie server **using a movie trailer identifier**. Thus, *Szeto* fails to cure the deficiency of *Rao et al.* and DTD in failing to disclose or suggest “**automatically identifying an executable using the content type determined from the metadata**,” as claimed.

At page 44 of the Appeal Brief, Applicants argued that *Szeto* is not properly applied in combination with *Rao et al.* and DTD because the IM application in *Szeto* is the actual data to be executed, and not an “executable,” as claimed. As was explained therein, the movie trailer in *Szeto* is the IM application, which is executed by a supporting application, e.g., a media player. As such, the “supporting application,” and not the movie trailer, is the “executable” in *Szeto*.

In response, at pages 4-5 of the Office Action, the Examiner asserted that because the IM application in *Szeto* is software for implementing an instruction set, and the supporting application is not a required application, since it is needed only when the IM application is

unable to render the corresponding data, Applicants' argument is erroneous. Applicants disagree. In those situations where the supporting application is not needed in *Szeto*, there is simply no executable at all. Where needed, the supporting application is the executable. It does not negate Applicants' argument that the movie trailer in *Szeto* is the IM application, i.e., actual data, and the "supporting application," not the movie trailer, is the "executable" in *Szeto*.

Independent claim 1 also recites "receiving at an electronic device a command specifying execution of an unidentified executable on first data." Applicants disputed the Examiner's assertion that *Szeto* teaches an unidentified executable, as claimed, by offering a summary depicted in a drawing at page 45 of the Appeal Brief. At page 5 of the Office Action, the Examiner asserted that it is not clear where, in *Szeto*, there is a correspondence to Applicants' drawing in the Appeal Brief.

As shown in the drawing in the Appeal Brief, and explained therein, at pages 45-46, in comparing the "sending client" of both the instant invention and *Szeto*, the sending client in the embodiments of the instant invention is a service provider and the command transmitted from the sending client identifies data of an unknown data-type, along with instruction. In *Szeto*, however, the sending client is not a service provider, the sending client sends an IM message also identifying data but specifying the data-type and having no instructions. Moreover, in *Szeto*, the execution is commanded by the receiving client, as opposed to the execution being commanded by the sending client in the embodiments of the instant invention.

In response to the Examiner's confusion as to what, in the *Szeto* disclosure, corresponds to Applicants' summary, reference is made to col. 12, lines 49-53, for example, where it is described that it is the "IM environments" that evaluate messages received by handlers in IM clients and it is the "IM environments" that determine an appropriate action for user and IM

messages, referring to Fig. 11 and steps 1104 and 1106. Thus, this is the support for Applicants' argument that *Szeto* executes data identified in the IM message and that such **execution is commanded by the IM environment of the receiving client**, and not the IM message. Contrast this with the embodiments of the instant invention wherein commands are sent by a service provider (sending client) to specify execution of identified data within a mobile phone (receiving client).

Thus, differences between the instant claimed invention and the disclosure of *Szeto*, and a reason why even if *Szeto* is combinable with *Rao et al.* and DTD (which it is not), the instant claimed subject matter would not result, are that whereas the execution is commanded by the receiving client in *Szeto*, the execution is commanded by the sending client in the instant claimed invention, and whereas the execution is identified by the sending client in *Szeto*, the execution is identified at the receiving client in the instant claimed invention.

At pages 5-6 of the Office Action, the Examiner disagreed, contending that in *Szeto*, execution is commanded by the sending client "because the IM message formed by and received from the sending client (which is also in line with applicants' analysis of *Szeto*, wherein the sending client providing the receiving client with information about an IM application), wherein based on what the IM message is, as the IM environment evaluates the IM message, an appropriate action is determined." Respectfully, the Examiner's analysis is flawed. The IM message sent by the sending client in *Szeto* identifies certain data and data-type but has no instructions associated therewith. Since there are no instructions, the execution in *Szeto* cannot reasonably be said to be commanded by the sending client. In *Szeto*, it is only after the instructionless IM message is received by the receiving client that the identified data in the IM message is retrieved and an executable is selected, at the receiving client, and an IM environment

decides what action to take. While that action may be based on data received from the sending client, it is not reasonable to conclude, as the Examiner has done, that this means that the sending client actually “commands” the execution. It does not.

At page 6 of the Office Action, the Examiner disputed Applicants’ argument that the combination of references does not teach or suggest an executable being determined at the receiving client, asserting that *Szeto* teaches “the receiving client utilizing an identifier to determine the executable (e.g. IM application or supporting application)...IM application and supporting application are both software for rendering data, and not data to be rendered by an application.” As explained above, the IM application in *Szeto* is **actual** data. It is the supporting application that would correspond to the “**executable**.” But the supporting application, if any, in *Szeto* is sent with information about the IM application by the sending client to the receiving client. Thus, the supporting application (executable) is already determined at the sending client prior to sending to the receiving client. Accordingly, unlike the instant claimed subject matter, the executable in *Szeto* is determined at the sending client, and not at the receiving client.

At pages 6-7 of the Office Action, the Examiner disputed Applicants’ argument that the proposed combination of references fails to teach or suggest the claimed feature of “metadata” and “the content type determined at the receiving client” because *Szeto* does not teach metadata and its content type is specified by the sending client. In particular, the Examiner asserted that Applicants are arguing the references individually. Applicants respectfully disagree.

It is not arguing the references individually when an applicant argues that a specific feature for which a particular reference is employed is, in fact, non-existent in that reference. Accordingly, it is not arguing the references individually to argue that *Szeto* fails to suggest

determining content type at the receiving client when the Examiner employs *Szeto* for that feature. *Szeto* clearly teaches determining content type at the sending client because that information is already known when the sending client transmits the information to the receiving client. The type of content is not determined at the receiving client in *Szeto* (See col. 13, lines 2-10).

In response to Applicants' argument that the proposed combination of references does not teach or suggest the claim feature of "automatically determining, from metadata of the first data, a content type of the first data" at the receiving client, because *Szeto*'s first data requires retrieval from a server and is not presented in the IM message, the Examiner asserted, at pages 7-8 of the Office action, that *Szeto*'s first data (IM message) has an identifier, alleged to be "fully equivalent to the claimed metadata," and is utilized for determining the corresponding executable (IM application or supporting application) for rendering the received message.

As explained above, the IM message in *Szeto* does not command an executable. It merely provides data for an IM environment to determine an executable at the receiving client.. But, whereas the instant claimed subject matter requires a determination of "content type of the first data" at the receiving client, the content type is already determined at the sending client in *Szeto*, prior to transmission of the IM message to the receiving client. Therefore, in *Szeto*, the "content type of the first data" is not determined at the receiving client.

At pages 8-9 of the Office Action, the Examiner disputed Applicants' argument that the proposed combination of references fails to teach or suggest the claim feature of "automatically identifying an executable using the content type determined from the metadata." In particular, the Examiner, again, argues that Applicants are arguing references individually. Again,

reference to the Appeal Brief will show that Applicants are, indeed, arguing the combination of references but must, of necessity, address separate references for what they are alleged to teach.

Moreover, the Examiner asserted that *Szeto* teaches the determination of a suitable executable (IM application or supporting application) to be used by reading the properties of the identifier of the first data (IM message) at the receiving client.

As Applicants keep repeating, for the reasons above, the IM application of *Szeto* is not an executable. It would correspond to actual data, whereas the supporting application would correspond to an executable in *Szeto*. In *Szeto*, the executable, i.e., the supporting application, is not determined at the receiving client. Rather, in *Szeto*, the receiving client is told the content type, and hence, the executable required, by the sending client. The determination of the required executable is made at the sending client in *Szeto*. Therefore, there is no need for a **determination of an executable at the receiving client** in *Szeto*.

At page 49 of the Appeal Brief, Applicants argued that the problem solved by *Szeto* is how to control the information received in an IM environment and that *Szeto* teaches the control of information at the receiving client using an IM environment, thus teaching away from the instant claimed subject matter by using the receiving client to determine whether and what to do with received messages. The instant claimed invention allows the sending client to control whether processing of a command message at the receiving client should occur and allows the receiving client to determine what executable should be used. The Examiner responded, at pages 9-10 of the Office Action, by asserting that even though *Szeto* solves a different problem, it is still pertinent to Applicants' invention and does not teach away "as the sending client control whether processing of a message at the receiving client should occur and allow the receiving client to determine what executable should be used..., as the sending client's IM message include

the identifier to control the receiving client for determining what executable should be used.” Respectfully, the Examiner’s rationale is flawed. As argued above, *Szeto* clearly does not allow the receiving client to determine what executable should be used. Rather, the **receiving client is told what executable is to be used** at the receiving client via the specification, at the sending client, of the supporting application.

At pages 10-11 of the Office Action, the Examiner responds to Applicants’ argument of no cogent rationale for making the proposed combination by asserting that *Rao et al.* teaches a mobile handset utilizing Sych ML protocol including metadata for execution of a XML command, DTD teaches metadata indicating a content type where the indication conforms to a XML document Type Definition (DTD), and *Szeto* teaches execution of an unidentified executable utilizing XML protocol. Therefore, concluded the Examiner, it would have been obvious to include DTD’s metadata indicative of content type and *Szeto*’s execution of the unidentified executable into *Rao et al.*’s metadata and XML command, respectively.

Applicants refer to the Appeal Brief for a detailed explanation of why the proposed combination is improper. But, suffice it to say that since none of the three combined references discloses or suggests either “receiving at an electronic device a command specifying execution of an unidentified executable on first data” or “automatically identifying an executable using the content type determined from the metadata,” as in claim 1, for example, for the reasons in the Appeal Brief and above, the combination of *Rao et al.*, DTD, and *Szeto* fails to present a *prima facie* case of obviousness regarding the instant claimed subject matter.

Finally, at pages 11-12 of the Office Action, the Examiner responded to Applicants’ argument that *Szeto* is not pertinent the particular problem with which Applicants were concerned by asserting that Applicants are arguing the references individually. An argument

that a reference is not pertinent to the particular problem with which Applicants were concerned is, by its very nature, an argument against that reference individually. However, it is also an argument against the proposed combination of references because a reference that is not pertinent cannot properly be made part of a proposed combination as one of ordinary skill in the art would not have sought to combine a non-pertinent reference except through impermissible hindsight.

Moreover, the Examiner asserted that *Szeto*'s identifier is functionally equivalent to the claimed metadata for identifying an executable to operate on the received data, thus making *Szeto*'s identifier pertinent to the particular problem facing Applicants "because applicant's command specifying execution of an unidentified executable on first data is accomplished via the meta-data/identifier to identify the executable for operating on the first data." Applicants disagree.

Szeto does not identify the corresponding application, i.e., the supporting application (corresponding to the executable) using metadata as in DTD. Rather, *Szeto* identifies the supporting application (corresponding to the executable) using an identifier specified by a sending IM client. Therefore, *Szeto* is not concerned with the problem with which Applicants were concerned because Applicants were concerned with using a non-specific command sent from a device to perform a common process on a plurality of target devices without specific adaptation for each device. But, in *Szeto*, the receiving client decides whether and what to do with the identified data, rather than the sending client telling the receiving client whether and what to do with the identified data. Accordingly, *Szeto* teaches away from the instant claimed subject matter and is not properly combinable with the other two references.

Accordingly, the Examiner is respectfully requested to withdraw the rejection of claims 1, 3-10, 12-16, and 41 as obvious under 35 U.S.C. § 103(a).

Therefore, the present application, as amended, overcomes the rejections of record and is in condition for allowance. Favorable consideration is respectfully requested. If any unresolved issues remain, it is respectfully requested that the Examiner telephone the undersigned attorney at (703) 519-9952 so that such issues may be resolved as expeditiously as possible.

To the extent necessary, a petition for an extension of time under 37 C.F.R. § 1.136 is hereby made. Please charge any shortage in fees due in connection with the filing of this paper, including extension of time fees, to Deposit Account 504213 and please credit any excess fees to such deposit account.

Respectfully Submitted,

DITTHAVONG MORI & STEINER, P.C.

March 30, 2010
Date

/Phouphanomketh Ditthavong/
Phouphanomketh Ditthavong
Attorney/Agent for Applicant(s)
Reg. No. 44658

Errol A. Krass
Attorney for Applicant(s)
Reg. No. 60090

918 Prince Street
Alexandria, VA 22314
Tel. (703) 519-9952
Fax (703) 519-9958